Word Count:955

Video, Working with Relays

In this video, we will learn the basics of relays, their applications, working principle, and later we will learn how to interface the Relay to your Raspberry Pi, and finally we will do an activity with all the things we have learned so far in this section.

A relay is an electromechanical device that is similar to a switch. Many relays use an electromagnet to operate a switch mechanically, but other operating principles are also used, such as solid-state relays. You might be wondering why to use a relay when all it does is the same function as a switch.

The main operation of a relay comes in places where only a low-power signal can be used to control a high power circuit. The relays are thus used in high current applications to isolate the low power control circuit form the high power driver circuits. Thus, knowing about relays is vital if you want to do industrial or home automation.

So, let's see how a relay works.

The diagram shows an inner section diagram of a relay. An iron core is surrounded by a control coil. When the current starts flowing through the control coil, the electromagnet starts energizing and thus intensifies the magnetic field. Thus, the upper contact arm starts to be attracted to the lower fixed arm and thus closes the contacts completing the circuit. On the other hand, if the relay was already de-energized when the contacts were closed, then the contacts move in the opposite direction and make an open circuit. As soon as the coil current is off, the movable armature will be returned by the force of the spring.

Relays have the exact working of a switch. So, the same concept is also applied. A relay is said to switch one or more poles. Each pole has contacts that can be thrown in mainly two ways. They are

Normally Open Contact , also known as NO contact . This is also called a make contact. It closes the circuit when the relay is activated. It disconnects the circuit when the relay is inactive.

Normally Closed Contact also known as NC contact . This is also known as break contact. This is the opposite of the NO contact. When the relay is activated, the circuit disconnects. When the relay is deactivated, the circuit connects.

Along with the NC and NO contacts of a relay, we also have a common contact called COM.

You can see a two-channel relay module here. Two-channel relay module is just a circuit in which two separate relays are provided. This allows controlling two devices at a time. Thus, the more the number of channels, the more the number of devices we can connect.

Now let’s interface the Pi 4 with the 2 channel relay. The 2 channel relay has 4 pins to interface with raspberry pi. VCC, Ground, IN1, and IN2. The relay module needs 5 Volts input at the VCC pin and more than 30 mA of current to work. This is why we need an external power supply to provide the obvious deficit in the current. Here we will use a popular external power supply board called the MB102 Breadboard Power Supply. To know more about this power supply and why we used it, please check out the links in the resources section. If you want to work with Relays without using the external power supply, you need to buy a 3.3 V Relay or a Solid State Relay.

Connect the 5 Volt pin of the Power supply to the VCC pin of the relay module. Then Connect the GND pin of the Power supply to the GND pin of the Raspberry Pi 4. Now we will connect a jumper wire from another GND pin of the power supply to another GND pin of the Raspberry Pi. So, now the Pi, the relay module, and the power supply all share the same ground pin. Finally, interface a Push button across the GPIO pin 2, which will be used to control the relay. Now let’s connect some high-power device across the relay output. We are connecting a CFL across the NC and COM port, as shown in this circuit diagram.

GPIO Zero 1.5 version doesn’t have an inbuilt class for working with a Relay, like LDR or PIR. Thus, we have to use the OutputDevice Class to work with the relay. Open the RELAY BUTTON.py file in Thonny IDE and run the script. When you press the push button, you can see that the CFL becomes OFF. In the code, we first imported the button and OutputDevice class from the guizero library, then we created instances of these classes along with the pin number related to each class. The next set of codes are pretty self-explanatory.

I have an activity for you, with the concepts we have learned till now. I want you to create a smart light for your backyard with the Raspberry Pi 4. The Project will have an LDR, a PIR, and a Relay. This is how it's going to work. The LDR checks whether its night or daytime by the light intensity and only if its night and if it then detects a motion, will make the Relay to switch on a CFL in your backyard. I hope you are successful with the project. If you are stuck, you can check out the resources section for the code.

Summary

In this video, we have covered the following

● Basics Of Relay

● Applications of Relays

● Working Principle of a Relay

● Interfacing & Working with a Relay using Raspberry Pi

In the next video, we will learn to interface and work with the Ultrasonic Sensor and Motors.